Cloud Message Queue

What's CMQ

Product Introduction

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

What's CMQ
Overview

Last updated：2018-09-06 14:41:31

About CMQ Products

CMQ (Cloud Message Queue) is a distributed message system developed by Tencent Cloud based on open source message engines and such system is compatible with the AMQP protocol. When a message is produced, 3 copies of this message will be stored on the disk in real time, and the distributed Raft algorithm can ensure this message and its copies strictly consistent, thus providing message publication and subscription service, message rewind service, 1-to-N message delivery service, sequential message service and other services. It is featured with high reliability, high availability, cross-IDC, transparent dynamic scaling, messages related to production and consumption and other advantages. CMQ has been developed and updated for over 7 years, providing support for asynchronous messaging services including WeChat, WeBank, QQ Show, mobile QQ and other large businesses in Tencent.

For now, CMQ, as a commercial product of Tencent Cloud, provides highly available message cloud service in many regions across the world, and the hardware facilities in data centers will be installed according to the high standard of IDC independently developed by Tencent. Multiple data centers will be deployed in a single region. With such deployment, even if the entire data center is not available, the message publication service can still be available. In Shenzhen and Shanghai Finance Zones, the dual active and multiple active deployment programs in the same city will be available to keep the key financial trading system safe and secure.

At present, CMQ provides access solutions for a number of protocols, including HTTPS, MQTT, AMQP, MQTT and so on. In addition, it is compatible with the client interfaces of open sources, such as RabbitMQ, RocketMQ and Kafka, and provides SDK access in multiple languages, such as php, java, python and others.

Project	CMQ (HTTPS access)	CMQ (TCP access)	CKafka	CMQ (MQTT access)

Project	CMQ (HTTPS access)	CMQ (TCP access)	CKafka	CMQ (MQTT access)
Scenario-oriented	Highly reliable financial-level message queues are applicable to decoupling between systems, asynchronous calls and other scenarios in the core architecture. The https access is available on Tencent Cloud and the SDK access in multiple languages is also supported.	The product features and client interfaces are completely compatible with open source products, including AMQP RabbitMQ and RocketMQ, and data can be rapidly migrated to the cloud with "no requirement" at all.	The CMQ team has already been a major contributor to the open source ecosystem, Kafka. After simply modifying the ip address, developers can migrate data to Tencent Cloud (supporting Version 0.8, 0.9, 0.10 and other mainstream versions). It provides more reliable, safer and OPS free enterprise-level Kafka message service for developers.	Because CMQ is specially designed for developers specialized in the smart home and industrial Internet of Things, it completely supports various protocols of the Internet of things including MQTT.
Highlights	Messages can be retained without any limit, financial-level messages can be expanded horizontally with high reliability, and messages can be stored on the disk in real time, thus enabling dual active disaster recovery capability across data centers and	It is based on CMQ storage engine. Real-time messages can be stored in the disk with high reliability. It is compatible with Version RabbitMQ 3.6.11 and Version RocketMQ 4.1.0. After the client of the	It is featured with ultra-high read/write performance. With the same equipment installed, its performance is over 2 times of the open source version of kafka. Compared with kafka, which is installed with purchased virtual machines, it has the following advantages, for	It provides the MQTT-based message publication/subscription mode, supports 1-to-N and 1-to-1 message interaction and provides TLS/SSL security encryption communications. In addition, it supports the access of a great number of clients at the million level for the persistent connection, the online and offline notification delivery, Last will and

Project	CMQ (HTTPS access)	CMQ (TCP access)	CKafka	CMQ (MQTT access)
	supporting advanced capabilities, such as sequential messages, dead letter queues, message rewind and message encryption.	native open source version is configured slightly, it can be connected to the cloud server.	example, saving 30~50% of the cost. OPS free, supporting horizontal expansion, unlimited offset accumulation, offering technical consultation service for upstream and downstream ecology, and providing rapid access, failure location, performance optimization and other services with regard to your real-time and offline computing platforms, such as Hadoop and Spark.	other highlighted features.

Note:

1. CMQ (TCP access) and CMQ (MQTT access) are under internal trials. You are most welcome to contact our after-sales engineers with your ticket and your personal engineer will be here to enable the internal trial service for you.

2. It supports the customized deployment. You are most welcome to contact our after-sales engineers with your ticket and your personal engineer will be here to enable the internal trial service for you.

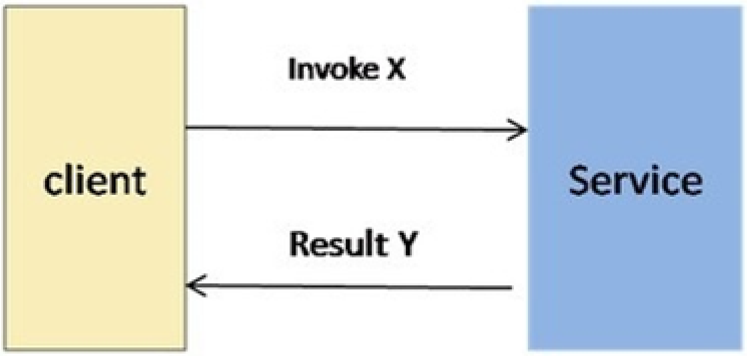
Product Overview

Tencent Cloud's Cloud Message Queue (hereinafter referred to as CMQ) is a distributed message queue service used for storing messages transmitted between processes. It is designed to provide reliable message-based asynchronous communication service between different applications deployed in a distributed way or between different components of an application. Messages are stored in a highly reliable and available message queue, which allows multiple processes to perform read and write operations simultaneously without interfering with each other. With Tencent Cloud CMQ, messages can be transmitted, without any data loss, between distributed components of applications executing different tasks. There is no need to keep every component available at all time.

The queue acts as a buffer between the data sender and the data receiver, eliminating the problems caused by the circumstances in which the data sender works faster than the data receiver or the data sender or receiver only connects to the network intermittently.

"Message" in CMQ indicates data transmitted between processes of the same computer/different computers while "message queue" is a container for retaining messages during the transmission. The message is sent to the queue, which acts as an intermediary to relay the message from its source to its destination.

The figure below shows an example of the traditional process communication mode, in which the client requests a service from the server and waits for a response from the server. However, such a mode has many shortcomings, for example, the likelihood of lost request in case of poor network conditions, or failed request due to timeout caused by overlong waiting of client in case of lengthy processing of server.



Typical Invoke/Respond Model

To deal with these problems, Tencent Cloud introduces the CMQ service for message distribution and management. With Tencent Cloud CMQ, components of an application can be separated to run independently, and message management between the components can be simplified. Any component of a distributed application can store messages in the queue, and Tencent Cloud CMQ ensures that each message is transmitted at least once and can be read and written many times. A single queue can be used by multiple distributed application components at the same time which are not required to collaborate with each other. All the components can programmatically retrieve and operate messages using CMQ APIs.

Overview of Application Scenarios

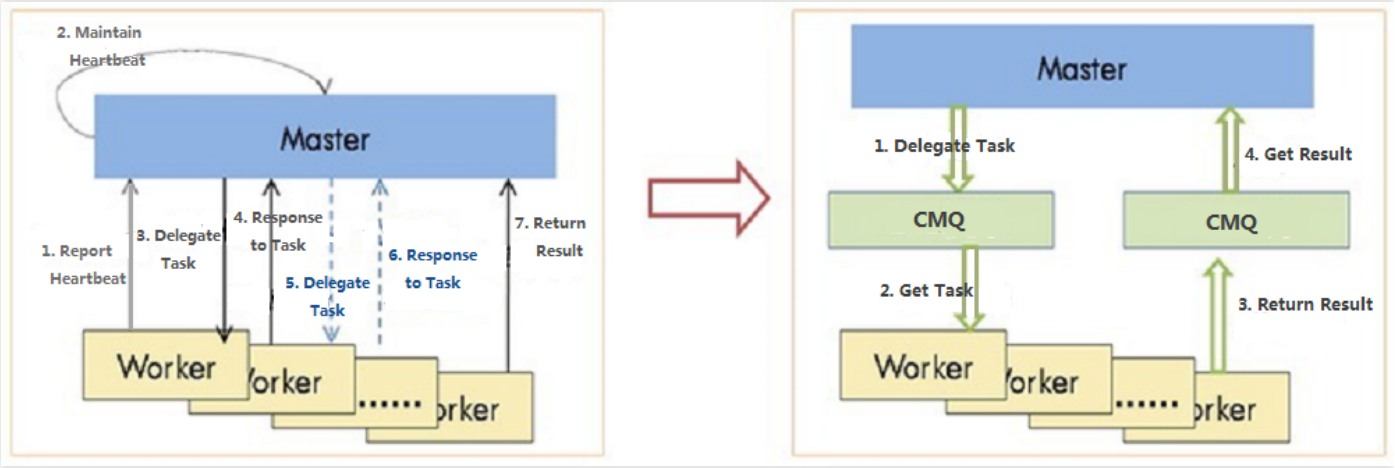
CMQ is recommended for application scenarios in which asynchronous communication is required. We take the following examples for illustration.

- In applications where reliable delivery of a message needs to be ensured, even if the receiver is unavailable for reasons such as power outage, crash or CPU overload when the message is being sent from the sender, receiver can still receive the message once becoming available. This function is not available in traditional message queues since it only stores messages in memory. Tencent Cloud CMQ enables messages to be stored persistently until the message is acquired by the receiver successfully.

- A message queue should run normally even if the number of messages stored in the queue and visits keep increasing. The traditional message queue stores messages in local memory. Since the processing capability of stand-alone computers and memory capacity are limited, the queue is not scalable. In contrast, Tencent Cloud CMQ, featuring distributed architecture, can ensure the simple scalability of the queue and the scalability is completely visible to CMQ users.

- Two services need to communicate with each other in the case that the networks cannot be interconnected or the routing information (such as IP and port) of applications is not known. For example, if two services on Tencent Cloud need to communicate while they do not know each other's address, the communication can be achieved in such a way that one party sends messages to and the other party receives the messages from the queue agreed upon by both parties.

- Communication between system components or applications is frequent so that the components or applications need to maintain their network connections, and there are multiple communication contents. In such a scenario, traditional architecture will complicate the system design. For example, when a central processing service needs to assign tasks to multiple task processing services (similar to the master-worker mode), the master needs to maintain connections with all the workers and determine whether the workers have started to process the tasks so as to decide whether to reassign tasks. At the same time, the workers need to report task processing results to the master. Maintaining such a system at the same level will complicate the design, thus increasing the implementation difficulty and maintenance cost. As shown below, Tencent Cloud CMQ can reduce the coupling between the two sides, making the system simpler and more efficient.



- The coupling between system components or applications is tight and needs to be reduced, especially when the controllability over its components is relatively weak. For example, when a company's business CGI receives user submissions, it will store some data in its own system and forward the processed data to other business applications (such as data analysis system, data storage system, etc.). The traditional solution is to establish a connection between services through the socket. In this way, if the receiver's IP or port changes, or the receiver changes, the data sender needs to modify the relevant information accordingly. But with Tencent Cloud CMQ, the sender and the receiver do not know each other's information, thus greatly reducing the coupling between them.

Product Features

Last updated：2017-12-19 15:44:23

Asynchronous Communication Protocol

After sending a message to the message queue, the sender can go back immediately instead of waiting for a response from the receiver. The message will be retained in the queue until it is fetched by the receiver. Therefore, the message is sent and processed asynchronously.

Improved Reliability

In traditional mode, message requests may fail due to overlong waiting. But in CMQ mode, if the receiver is unavailable when a message is sent, the message will be retained in the queue until it is successfully delivered.

Process Decoupling

CMQ can reduce the coupling between two processes. As long as the message format remains unchanged, the sender needs no modification even if the receiver's API, location or configuration are changed. Besides, the sender doesn't need to know the receiver when sending messages, thus simplifying the system design. However, in traditional mode, processes are connected via a remote procedure call (RPC) or the socket. When the API, IP or port of one side changes, the other must modify request configurations accordingly.

Message Routing

The sender needs no a direct connection with the receiver since CMQ allows messages to be routed from the sender to the receiver. Even for two services between which network interworking is difficult to be available, the message routing also works.

Multi-terminals

Multiple parts of the user system can send or receive messages simultaneously, and Tencent Cloud CMQ controls message availability through message status.

Diversity

All queues can be configured independently and can be different. You can customize queue configurations for different application scenarios. For example, if the message processing in a queue takes a long time, you can optimize queue attributes.

Product Advantages

Last updated：2018-05-23 14:45:52

Compared with various open source message queue products, CMQ has the following advantages:

Advantages over Kafka

1. **Kafka is not suitable for core financial business:** CMQ features synchronous flush mechanism, ensuring reliable generation of data. If messages generated on the client are flushed to the disk by over half of brokers in "set", the acknowledgement will be returned to notify that messages are generated successfully. If the client does not receive the acknowledgement within a certain period, the messages will be resent to ensure that messages are sent successfully. However, Kafka features asynchronous flush, that is, asynchronous replication, which may lead to message lost. Therefore, Kafka cannot ensure reliable delivery of messages.

2. **If a consumption failed, Kafka cannot try it again:** For example, a top-up failure may be caused by high pressures on the operator gateway called by a top-up application; but if the application recalls the gateway later, the consumption may succeed. This is also required when deducting money from bank accounts using Alipay. The retry needs to be reliable, which means a failure and retry message will not be lost if a consumer crashes. CMQ can retry the consumption several times.

3. **Compared with the message trace by CMQ, locating problem is cumbersome in Kafka:** Message trace refers to the complete link information such as when and where the producer sends the message to the consumer and the consumer consumes the message. It is used to quickly locate the problems in financial businesses.

Advantages over RabbitMQ

1. **Higher QPS in CMQ:** In highly reliable scenarios, the throughput of CMQ is over four times larger than that of RabbitMQ in equivalent physical devices. The QPS of a single cluster is over 100,000.

2. **RabbitMQ does not support message rewind:** RabbitMQ does not support message rewind. But CMQ can rewind messages by time. For example, CMQ can re-consume messages sent after a certain time point of the previous day. A typical application scenario for message rewind is the order analysis of a consumer. For example, if all messages consumed today become invalid due to failures of program

logic or dependent system, the messages sent after 00:00 of the previous day need to be re-consumed. At this time, time-oriented message playback is very helpful for business recovery

3. **Comparison of consistency algorithms:** Although CMQ and RabbitMQ both support hot backup using multiple servers for high availability, the Raft algorithm- based CMQ is easy to understand and maintain, while the Guaranteed Multicast-based RabbitMQ is hard to learn.

4. **Great difficulty with RabbitMQ O&M:** The programming language of RabbitMQ is Erlang, which is not common and costs more to study.

Advantages over RocketMQ

1. **Data may be lost in RocketMQ in extreme cases:** RocketMQ can return acknowledgment to the client when flush is not made. In this way, messages will be lost if the server crashes.

2. **Multiple masters and slaves are required in RocketMQ to ensure highly availability of services:** When there is no surviving node in ISR, RocketMQ cannot ensure the availability and reliability, leading to higher cost.

Therefore, compared with traditional open source MQ applications, Tencent Cloud CMQ has the following advantages:

	Tencent Cloud CMQ	Open Source Message Broker
High Performance	Featuring both high performance and high reliability. The QPS of a single CMQ instance is up to 5,000	Featuring either high data reliability or high performance

	Tencent Cloud CMQ	Open Source Message Broker
High Scalability	High scalability in number of queues and queue storage capacity The underlying system can perform auto scaling according to the business scale and the upper layer service will not be affected CMQ can efficiently send, receive, push and retain over hundreds of millions messages. There is no upper limit to the capacity Services are available in multiple regions including Beijing, Shanghai and Guangzhou	Limited number of queues and retained messages Each IDC must be redeployed with the purchased devices, which is very complex
High Reliability	Based on CRMQ (Cloud Reliable Message Queue), the distributed architecture independently developed by Tencent Cloud, CMQ has been widely applied in Tencent's services such as Red Packet in QQ and WeChat and lottery When each message is returned to users and written, the message service ensures that 3 copies of the data has been generated and written into different physical machines. In addition, the backend data replication mechanism provides rapid data migration when any physical machine fails and guarantees that there are always three copies of user data available. The reliability is up to 99.999999% The improved Raft consistency algorithm is used to ensure strong consistency of data Guaranteed service availability: 99.95%	Data is stored on a single server or servers with simple master-slave structure. In this way, data SPOF problem may occur so that data cannot be rewound once lost Open source Replicia algorithm is used. Adding or deleting a server node will cause the rebalancing of global data, leading to a sharp decline in availability For example, Kafka cannot guarantee strong consistency of data by using asynchronous flush (asynchronous Replication)

	Tencent Cloud CMQ	Open Source Message Broker
Business Security	Multi-dimension security protection and anti-DDoS attack services Each message service is provided with a separate namespace. The data of different clients are strictly isolated Support HTTPS access Support cross-region security messaging service	Limited security protection Cross-region and cross-IDC public network services are not always provided to protect from threats from public network